/*Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of integers (Array Implementation of Queue with maximum size MAX)
a. Insert an Element on to Circular QUEUE
b. Delete an Element from Circular QUEUE
c. Demonstrate Overflow and Underflow situations on Circular QUEUE
d. Display the status of Circular QUEUE
e. Exit
Support the program with appropriate functions for each of the above operations.*/

```
#include <stdio.h>
#include<stdlib.h>
#define MAX 3
int cq[MAX];
int front = -1,rear = -1;
void inset(int);
void delete();
void display();
int main()
{
 int ch;
 int item;
 while(1)
 {
 printf("\n\n~~Main Menu~~");
 printf("\n==> 1. Insertion and Overflow Demo");
 printf("\n==> 2. Deletion and Underflow Demo");
 printf("\n==> 3. Display");
 printf("\n==> 4. Exit");
 printf("\nEnter Your Choice: ");
 scanf("%d", &ch);
 switch(ch)
 {
 case 1: printf("\n\nEnter the element to be inserted: ");
 scanf("%d", &item);
 insert(item);
 break;
 case 2: delete();
 break;
 case 3: display();
 break;
 case 4: exit(0);
 default: printf("\n\nPlease enter a valid choice");
 }
 }
}
void insert(int item)
{
 if(front ==(rear+1)%MAX)
 {
 printf("\n\n~~Circular Queue Overflow~~");
 }
 else
 {
 if(front == -1)
```

```
 front = rear = 0;
else
 rear = (rear+1)%MAX;
 cq[rear] = item;
 }
}
void delete()
{
 int item;
 if(front == -1)
 {
 printf("\n\n~~Circular Queue Underflow~~");
 }
 else
 {
 item = cq[front];
 if(front == rear) //only one element
 front = rear = -1;
 else
 front = (front+1)%MAX;
 printf("\n\nDeleted element from the queue is: %d ", item );
 }
}
void display()
{
 int i ;
 if(front ==-1)
 {
 printf("\n\nCircular Queue Empty");
 return;
 }
 else
 {
 printf("\nCircular Queue contents are:\n");
 printf("\nFront[%d]-> ", front);
 for(i=front; i!=rear ; i=(i+1)%MAX)
 {
 printf(" %d", cq[i]);
 }
 printf(" %d", cq[i]);
 printf(" <-[%d]Rear", rear);
 printf("\n");
}
}
```